ISO TC184/SC4/* WG6 N101 (P__)

*Complete with EC (for editing), PMAG, or WG

Date: October 11, 1995 **Supersedes SC4/ WG6 N 97 (P__)**

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: 35 **Title:** Conformance Testing Methodology and Framework: Abstract Test Methods for SDAI Implementations

Purpose of this document as it relates to the target document is:

x Primary Content
_ Issue Discussion
_ Alternate Proposal
_ Partial Content

Current Status: working

ABSTRACT:

This part of ISO 10303 describes test methods for use in the conformance testing of software systems implementing, in one or more language binding, a level of functionality (implementation class) as specified in ISO 10303-22.

KEYWORDS:

SDAI testing, test purpose, test suite, language binding, late binding, early binding,, Data Access Interface

Document Status/Dates

Part Documents	**Other SC4 Documents**
____ Working Draft	____ Working
____ Project Draft	____ Released
____ Released Draft	____ Confirmed
____ Technically Complete	____ Approved
____ Editorially Complete	
____ ISO Committee Draft	

Owner/Editor: Shantanu Dhar
Address: Industrial Technology Institute, 2901 Hubbard Road, Ann Arbor, MI 48105, USA
Telephone: +1 313 769 4381
FAX: +1 313 769 4064
E-Mail: sxd@iti.org

Alternate:
Address:
Telephone:
FAX:
E-Mail:

Comments to Reader

This is an initial version of Part 35 and incorporates comments from the Washington DC ISO meeting. It includes an example abstract test case in Annex A.

this page intentionally blank

Contents

Foreword

The ISO (International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has a right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303-22 is being prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC4, *Industrial data and global manufacturing languages.*

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration - Product data representation and exchange:*

– Part 1, Overview and fundamental principles;

– Part 11, Descriptive methods: The EXPRESS language reference manual;

– Part 21, Implementation methods: Clear text encoding of the exchange structure;

– Part 22, Implementation methods: Standard data access interface;

– Part 31, Conformance testing methodology and framework: General concepts;

– Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients[1)];

– Part 33, Conformance testing methodology and framework: Abstract test cases;

– Part 41, Integrated generic resources: Fundamentals of product description and support;

– Part 42, Integrated generic resources: Geometric and topological representation;

– Part 43, Integrated generic resources: Representation structures;

– Part 44, Integrated generic resources: Product structure configuration;

– Part 45, Integrated generic resources: Materials;

– Part 46, Integrated generic resources: Visual presentation;

– Part 47, Integrated generic resources: Shape variation tolerances;

– Part 49, Integrated generic resources: Process structure and properties;

---

[1]To be published.

- Part 101, Integrated application resources: Draughting;
- Part 104, Integrated application resources: Finite element analysis;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 210, Application protocol: Printed circuit assembly product design data;
- Part 213, Application protocol: Numerical control process plans for machined parts.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Part 11 specifies the description method;
- Part s 21 to 26 specify the implementation methods and language bindings;
- Parts 31 to 35 specify the conformance testing methodology and framework;
- Parts 41 to 49 specify the integrated generic resources;
- Parts 101 to 105 specify the integrated application resources;
- Parts 201 to 213 specify the application protocols.

Should further parts be published, they will follow the same numbering pattern.

The reader may obtain information on the other Parts of ISO 10303 from the ISO Central Secretariat.

Annex A forms an integral part of this document.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

ISO 10303 is organized as a series of parts, each published separately. The parts of this International Standard fall into one of the following series: descriptive methods, integrated resources, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the conformance testing series.

This part of ISO 10303 specifies test methods used in the conformance testing of software systems implementing the Standard data access interface (SDAI) for one or more language bindings in one of the classes of implementation specified in ISO 10303-22.

Industrial automation systems and integration - Product data representation and exchange - Part 35: Conformance Testing Methodology and Framework: Abstract Test Methods for SDAI implementations

1. Scope

This part of ISO 10303 presents the abstract test methods and requirements for conformance testing of an implementation of a language binding of the SDAI. Since the SDAI is specified independently of any programming language, the abstract test methods presented in this part will be applicable to all language bindings specified for the SDAI. The abstract test methods specified herein will also address early and late bindings to application protocol schemata as well as the various classes of implementation detailed in ISO 10303-22.

The following are within the scope of this document:

– abstract test methods for software systems that implement the SDAI;

– conformance tests for all SDAI operations specified in ISO 10303-22;

– the specification, in a manner that is independent of any binding language[1], of algorithms and approaches for the testing of various SDAI operations;

– the specification and documentation of abstract test cases

The following are outside the scope of this document:

– the development of test data and/or test programs for specific language bindings;

– the specification of test methods, algorithms, or programs for the conformance testing of applications that interact with SDAI implementations;

– implementation policy for a conformance test system that realizes the test methods specified in this part of ISO 10303.

2. Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 10303-1	*Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.*
ISO 10303-11	*Industrial automation systems and integration - Product data representation and exchange - Part 11: Descriptive methods: The EXPRESS language*

[1]Binding-specific functions need to be tested as well. Although this part of ISO 10303 in language-independent, the issue of testing such functions needs to be addressed.

	reference manual.
ISO 10303-21	*Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure.*
ISO 10303-22	*Industrial automation systems and integration - Product data representation and exchange - Part 22: Implementation methods: Standard data access interface.*
ISO 10303-31	*Industrial automation systems and integration - Product data representation and exchange - Part 31: Conformance testing methodology and framework: General concepts.*
ISO 10303-32	*Industrial automation systems and integration - Product data representation and exchange - Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients.*
ISO 10303-33	*Industrial automation systems and integration - Product data representation and exchange - Part 33: Conformance testing methodology and framework: Abstract test cases.*
ISO 10303-34	*Industrial automation systems and integration - Product data representation and exchange - Part 34: Conformance testing methodology and framework: Abstract test methods.*

3. Definitions

3.1. Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

- abstract test suite;
- application protocol;
- conformance class;
- implementation method;
- Protocol Implementation Conformance Statement;
- PICS proforma;

3.2. Terms defined in ISO 10303-22

This part of ISO 10303 makes use of the following terms defined in ISO 10303-22:

- application schema;
- implementation class;
- domain equivalent;

- language binding;
- repository;
- schema instance;
- SDAI-model;
- session;
- validation;

3.3. Terms defined in ISO 10303-31

This part of ISO 10303 makes use of the following terms defined in ISO 10303-31:

- abstract test case;
- abstract test method;
- conformance;
- conformance log;
- conformance report;
- conformance testing;
- executable test case;
- FAIL (verdict);
- implementation under test;
- INCONCLUSIVE (verdict);
- PASS (verdict);
- Protocol Implementation eXtra Information for Testing;
- PIXIT proforma;
- test campaign;
- test case error;
- test laboratory;
- test report;
- (test) verdict;
- verdict criteria;

3.4. Abbreviations

The following abbreviations are used in this part of ISO 10303:

IUT	Implementation Under Test
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation eXtra Information for Testing
SDAI	Standard Data Access Interface

4. Requirements and overview

A conformance test system for SDAI implementations will possess the capability of testing a SDAI implementation (the IUT) for its claimed behavior against requirements specified in ISO 10303-22 and companion parts depending on the programming language the IUT supports.

This part of ISO 10303 describes abstract test methods that conformance test systems would implement in order to test SDAI implementations. General principles and an overall framework for conformance testing are provided in ISO 10303-31. Requirements on test laboratories are defined in ISO 10303-32. The methods for preparing, controlling, observing and analyzing implementations during testing are defined in this part of ISO 10303.

4.1. SDAI Testing requirements

These test methods described in this part of ISO 10303 will address the following testing requirements:

– An SDAI implementation may be early bound, late bound, or both. Test methods specified in this part of ISO 10303 address all such implementations.

– An SDAI implementation must obey the state model described in ISO 10303-22. The test methods specified in this part ensure this is tested.

– ISO 10303-22 is written independent of any programming language. The test methods specified for SDAI will therefore be generic.

– ISO 10303-22 specifies various levels of functionality that an SDAI implementation can support. The test methods specified herein will facilitate the testing of implementations at these various levels or implementation classes.

– SDAI operations provide means for data manipulation and transactions. Testing these operations will provide assurance of their correctness and whether they had the desired effect on the persistent storage. These operations include create, delete, modify, validate and various manipulation operations that act on schema instances, SDAI-models, and instances of application schema entities.

– In situations of error, SDAI operations return error codes. Testing will encompass all reasonable error situations for an operation to ensure that appropriate error codes are returned.

– Dictionary information operations provide subtype/supertype and complex entity information, interface information, and domain equivalence information. In the case of late bound applications, the dictionary is *closed*[2], and will be tested implicitly.

– Environment and session operations provide the capability for changing the state of the SDAI session. Conformance testing will ascertain that only the permitted transitions take place and that only permitted operations for a state are allowed.

– The testing of error handling requires checking the returned error value against the actual error condition which triggered that return value.

– Aggregate operations play important roles in SDAI implementations, facilitating the successful completion of other complex operations or sequences of operations.

– Verdicts must be issued for all test cases executed.

– Test logs and reports that accurately capture the results of all test cases executed must be generated.

– Test campaigns with potentially large numbers of test cases will have to be managed, sequenced, and run.

4.2. SDAI implementation types

This clause describes the nature of SDAI implementations that are addressed by the test methods specified in this part. SDAI implementations can be of two types - implementations that accept requests encoded in the binding language of the implementation and respond to these requests in the manner required by the standard, and applications that make a series of these requests to achieve a meaningful objective.

– A SDAI implementation encapsulates repositories and presents an interface for the manipulation of these repositories and schema instances. This is the interface which applications would use to interact with SDAI implementations.

– Application programs use the SDAI interface to communicate with a SDAI implementation to perform their tasks. These tasks typically require access to repositories and schema instances. The SDAI implementation provides this access via the interface.

This part of ISO 10303 presents test methods for the testing of implementations of the former kind. Test methods for applications that access SDAI implementations are not presented in this part.

4.3. Testing architecture

The architecture assumed by the test methods for SDAI implementations is a simple *client-server*[3] relationship between the SDAI implementation (IUT) and the test engine. The test application, which is a part of the test engine, plays the role of an application by making SDAI operation requests to the IUT. The IUT returns output back to the test application. A simplified description is presented in Figure 1.

The test application processes the outputs and, in turn, communicates appropriate values to the test engine for recording.

[2]This needs to be described in more detail.

[3]The use of the term *client-server* has been viewed by some members of WG6 and 7 as carrying too much meaning, and thereby subject to misinterpretation by readers. It is still here since the editor of this part has not found a sufficiently high-quality substitute.

The test application uses test data as appropriate to execute its test cases.

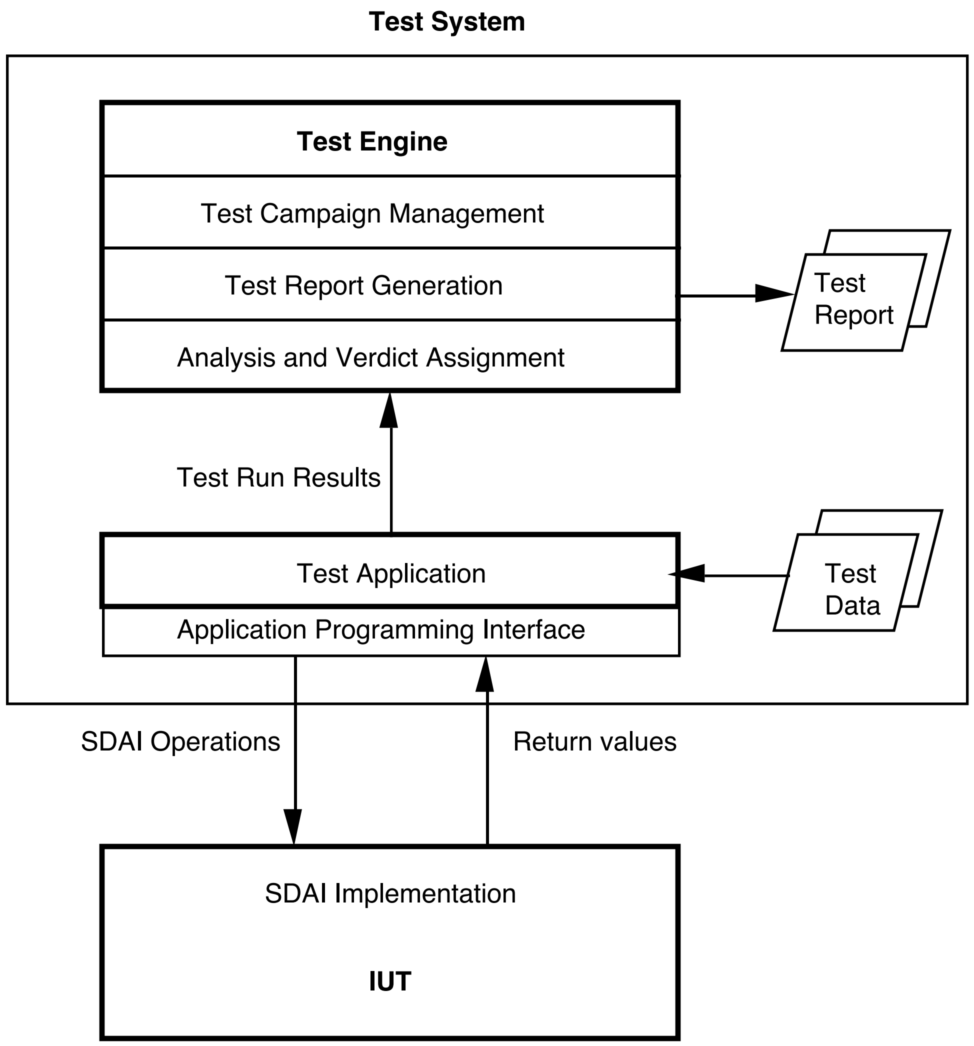

Figure 1 Testing architecture for SDAI implementations

5. Testing process

5.1. Preparation for testing

PICS and PIXIT proformas are completed by IUT vendors prior to testing. The information in these forms is used to guide test case selection, sequencing, and input data to be used in the execution of the test campaign.

Abstract test cases are built around SDAI operations specified in ISO 10303-22. Since there exists no guaranteed one-to-one relationship between operations in ISO 10303-22 and a particular language binding, executable test cases are developed to logically match these operations so that the executable test cases can be grouped according to implementation level.

> Note: This part of ISO 10303 may also need to specify detailed guidelines for developing and documenting abstract test cases.

Executable test cases are built from the abstract test cases using PICS and PIXIT information for an IUT. The sequence of their execution in a test campaign is determined at this stage.

> Note: In assembling executable tests for an IUT, of particular importance is information about the IUT's implementation class, whether it is early or late-bound, the application schema it is bound to (if early bound), and the programming language of choice.

For a further description of conformance testing preparations from the perspective of the testing laboratory, see ISO 10303-32.

5.2. Test campaign

A conformance test campaign is a sequenced execution of all required executable test cases (ETCs). The results of this sequenced execution determines conformance.

Modifications to the IUT are not permitted during a test campaign. Modifications to the ETCs or to the sequence of their execution is not permitted during a test campaign, except in the situation where the ETC is determined to be in error.

If an ETC is determined to be in error, a verdict of INCONCLUSIVE shall be assigned to its execution until the error is resolved and the test repeated.

5.3. Test conclusion

A test campaign may terminate for any reason. A normal termination of a test campaign occurs when all its executable tests have been run. A PASS verdict is assigned to a campaign if all the tests of the campaign have returned PASS verdicts and no violation of any ISO 10303 part is detected.

5.4. Test report production

A conformance test report is created after a test campaign terminates. As a minimum, this report shall contain information identifying the IUT and the capability that is being tested (late/early binding, SDAI implementation class(es) supported, binding language, AP or application schema information, etc.) and a detailed reports and verdicts on all the executable tests in the campaign including a list of those not executed, if any. Any relevant information on the testing environment will be included as well.

6. Test method - early-bound implementation

An early bound SDAI implementation provides an interface specific to the application schema to which it is bound at compile-time. The test application that will have to be written to test an early bound IUT will therefore depend on this application schema (typically an AP).

The set of early bound functions that need to be tested can be classified in two broad groups: those that are schema-specific and those that are schema-independent.

The schema-independent functions can be tested in an uniform manner regardless of the application schema that is bound to the IUT. Test cases for the schema-specific functions will have to be developed for each application schema that may be early-bound to an IUT. The testing of schema-specific functions will typically require the use of schema-independent functions. Therefore, an efficient and realistic approach to testing would combine the two rather than test them in separate groups.

> Note: Typically for an AP, a very large number of schema-specific functions will be generated. WG6/7 will have to take a decision on whether it is necessary to exercise all these functions, or whether, an appropriately selected subset would suffice.

7. Test method - late-bound implementation

A late bound SDAI implementation provides a single generic interface, which may be used with any application schema. Nothing more can be said about the implementation since the implementor has complete freedom in adopting an implementation approach. In an extreme case, a late-bound implementation for a particular application schema could be early-bound, with an interface that mimics the interface of a late-bound implementation.

Therefore, the testing of late-bound implementations would also have to be in the context of a specific application schema, usually an AP. Testing would progress in a manner similar to the testing of early-bound implementations[4].

[4]How would an implementation advertising a general late-binding capability be tested? Would it have to be tested for each application schema they becomes available? Hopefully not!

Annex A
(informative)
Example abstract test cases

Introduction

This is a first cut at writing test cases for SDAI. Comments are welcome.

Test cases for SDAI implementations, like those for the exchange structure would be driven by test purposes (TPs) and verdict criteria (VCs). A TP specifies explicitly a small testable element of a specification. For example, in the conformance testing of an implementation embodying an AP, a TP is associated with each entity in the AIM stating (in effect) that a conforming implementation must instantiate instances of the entity in all its allowable forms. VCs, on the other hand mirror TPs. A VC describes the nature of the characteristics in the output that would imply conformant behavior of the IUT. Often, more than one VC is needed to check the satisfaction of a TP by the IUT.

A suite of test cases in SDAI would exercise the operations specified in ISO 10303-22 while ensuring that all legal state transitions are exercised, and each error is flagged at least once.

A test case therefore consists of five components:

– One or more TPs which clearly specify the aspect of the specification being tested,

– Test data that would be used to perform the operations,

– The sequence of operations that would be required to cover the TP(s),

– Reference results against which the results of the test would be compared.

– VCs that provide the basis for comparison between the test results and the reference.

Example Test Case 1

Test Purpose

The `open repository` and `close repository` operations (ISO 10303-22 11.3.4) will be executed in accordance to their specifications. All possible error conditions will be returned at least once.

Test Data

None.

Note: Test cases that need test data would provide non-model data (like repository name, SDAI-model name, etc. in natural language, and model data in Part 21 format.

Sequence of Operations

1a) Ensure by earlier instantiation that a sdai_repository identified by `test_repo_1` will be available upon the execution of a `open session` operation.

b) Perform a `open session` operation, ensuring that it succeeds (i.e. no error indications are returned).

c) Perform a `open repository` operation, supplying as inputs `test_repo_1` and the session identifier or handle returned by the `open session` operation performed in b) above. Use implementation-specific capability to determine the contents and cardinality of the list **active_servers** of the current session.

d) Repeat 1c) above.

e) Perform a `close repository` operation, supplying as input `test_repo_1`. Use implementation-specific capability to determine the contents and cardinality of the list **active_servers** of the current session.

f) Repeat 1e) above.

g) Repeat 1c) above replacing `test_repo_1` with `test_repo_2`.

h) Perform a `close repository` operation, supplying as input `test_repo_2`.

i) Perform a `close session` operation.

j) Repeat 1c) above.

k) Repeat 1i) above.

Reference Results

None.

Verdict Criteria

1) Upon execution of 1c), the list **sesson.active_servers** will contain the single repository `test_repo_1`.

2) Upon execution of 1d), the operation will return the error indicator RP_OPN.

3) Upon execution of 1e), the list **sesson.active_servers** will be empty.

4) Upon execution of 1f), the operation will return the error indicator RP_NOPN.

5) Upon execution of 1g), the operation will return the error indicator RP_NEXS.

6) Upon execution of 1h), the operation will return the error indicator RP_NEXS.

7) Upon execution of 1i), the operation will return the error indicator SS_NOPN.

8) Upon execution of 1j), the operation will return the error indicator SS_NOPN.

Note 1: This example test case is not complete, since it does not test for all the error conditions of the two operations being exercised.

Note 2: This test case does not instantiate any data, i.e. it does not instantiate schema_instances, SDAI_models, or application_instances. Typically, such tests of environment or session operations could be combined with some data instantiation tests. However, test cases should have manageable levels of complexity and be constructed so that non-conforming elements can be clearly identified.

Annex B
(informative)
Issues Log

Each issue is set out as follows:

Issue number□hese are assigned sequentially

Issue date□he date given on the written issue or, where this is not given, the date on which the issue was received.

Author□he author of the issue may be an individual or a committee.

Status□he current status of the issue. This is categorized as follows:

Open□his issue has not been resolved by WG6

Resolved□ resolution to the issue has been agreed by WG6; a brief discussion of the resolution is given together with a reference to the version of the Part document in which the resolution is fully documented.

Rejected□he issue has been discussed by WG6 but has not been accepted. A brief statement of the reason for rejection of the issue is given.

Clarification sought□he issue is not understood by WG6; the author has been asked to supply additional information.

Deferred□he issue has been discussed by WG6 and, while it has been agreed that this issue is valid, it has been agreed that resolution will not be possible until a future version of the Part. A brief statement of the reasons for the deferral are given.

Other projects□re named (WGn/Pm) if WG6 has found it necessary or expedient to consult with other projects or working groups in resolving, rejecting or deferring an issue.

Note: Where other projects are indicated for an **open** issue, this indicates that WG6 (or, in some cases, the document editor) has identified a requirement to consult with other projects or working groups.

Issue text□he text of the issue as documented by its author.

Technical discussion□ee comments above regarding the statement of reasons for the acceptance, rejection or deferral of an issue.

Open Issues

Issue 1

Date: 12 June 1995

Author: Shantanu Dhar

Status: Open

The Guidelines for the development of abstract test suites address the testing of preprocessors and postprocessors, i.e. only the exchange structure is addressed. The specification, development, and documentation of abstract test cases and suites has to be addressed. Part 35 or a companion document would have to address this.

Technical Discussion: Currently under discussion.

Issue 2

Date: 12 June 1995

Author: Shantanu Dhar

Status: Open

WG 6 and 7 need to resolve the issue of exhaustive testing of early bound implementations. The question is: Is it necessary to exhaustively test every function for the creation, deletion, and modification of every entity type in the application schema? Or does a judiciously chosen representative set of such functions adequately address testing needs.

Technical Discussion: Currently under discussion.

Issue 3

Date: 12 June 1995

Author: Shantanu Dhar

Status: Open

ISO 10303-22 specifies error conditions for all operations. Testing will have to check whether an implementation does indeed return the appropriate error values when error conditions are encountered. This implies error test cases would have to be part of SDAI test suites.

Technical Discussion: Currently under discussion.

Index